# 1 - Voice Recognition: Introduction

This is a module on Voice Recognition for our ELEC 301 project.

## Introduction:

Automatic speech recognition, the conversion of spoken word to text, has posed a difficult problem for electrical engineers since the 1930s. Progress in the field is slow and difficult to measure due to the unpredictability of word error rates, which can vary anywhere from 1% to 50% depending on the scope of recognition and the requested task. Nevertheless, incremental improvements over the past thirty years and industrial applications have proven the usefulness of speech recognition software in a number of applications, including military communications, transcription of medical records, and the training of air traffic controllers. Applications for speech recognition systems continue to emerge, particularly in mobile devicing and video gaming, to which the recent inclusion of the Siri “intelligent personal assistant” to the Apple iPhone 4S may attest.

In this project, we investigate the basic implementations of speech recognition. We chose to restrain our goal to recognition of single-digit numbers from “0” to “9” for a hypothetical phone number recognition system.

## 2 - Voice Recognition: Overview of the Modern Algorithms

### Overview of the Modern Algorithm:

The speech recognition system consists of speech segmentation, feature extraction, and optimal feature-matching with a trained library of stored features.

### Speech Segmentation:

Speech segmentation is fairly uniform across systems, segmenting a string of spoken words into individual components. This can be easily accomplished by segmenting at points where power of the sampled signal goes to zero.

### Feature Extraction:

Feature extraction may be done in a variety of ways, depending on the features one chooses to extract. Industry standard is extraction of the coefficients that collectively represent the short-term power spectrum of the recorded sound, known as mel-frequency cepstrum coefficients (MFCCs). MFCCs are derived by:

1. Fourier transforming the windowed (usually Hamming) excerpt of the signal
2. Using triangular overlapping windows, map the powers of the obtained spectrum onto the mel scale, a logarithmic scale of pitches that more accurately models human hearing
3. Taking the log of the powers at each mel-frequency
4. De-correlate the resulting spectrum with a cosine transform
5. Extract the MFCCs as the amplitudes of the resulting spectrum

MFCC feature-extraction is typically used in conjunction with Hidden Markov Model feature-matching.

Prior to MFCCs, speech recognition systems used linear predictive coding (LPC). By assuming sibilants and plosive sounds to be occasional anomalies and therefore inverse-filtering out the formants, the values of the signal could be predicted on a local timescale by a series of linear representations after having extracted the coefficients.

Feature-matching:

Feature-matching is traditionally implemented via dynamic time warping (DTW), which allowing for the matching of sampled words with stored templates despite stretched and compressed differences in speed and timing. This technique has fallen out of favor thanks to the current industry gold standard of speech recognition: the Hidden Markov Model (HMM).

As speech signals are short-time stationary processes, modeling speech signals as HMMs is feasible - and offers great advantages over DTW due to extensive training features and implications towards a tremendously robust recognition system. The HMM-based approach is complex, but at the highest level involves the following:

1. Each word has been broken down into phonemes, its smallest linguistic segments, and the HMM output distribution for each phoneme trained and stored beforehand
2. The HMM outputs sequences of n-dimensional real-valued vectors consisting of MFCCs every few milliseconds
3. Each state of the HMM contains a statistical distribution of a mixture of diagonal covariance Gaussians that indicate the likelihood of each vector
4. The HMM for the targeted word is then identified by concatenating the stored HMMs per target phoneme

# 3 - Voice Recognition: Chosen Methods of Investigation

## Chosen Methods of Investigation

The limited timeframe of our project meant both DTW and HMM-based approaches were impractical, requiring many hundreds more man-hours than was available. We chose to focus on achieving solid results from a more primitive algorithm, the LPC, and work on making it more robust thereafter.

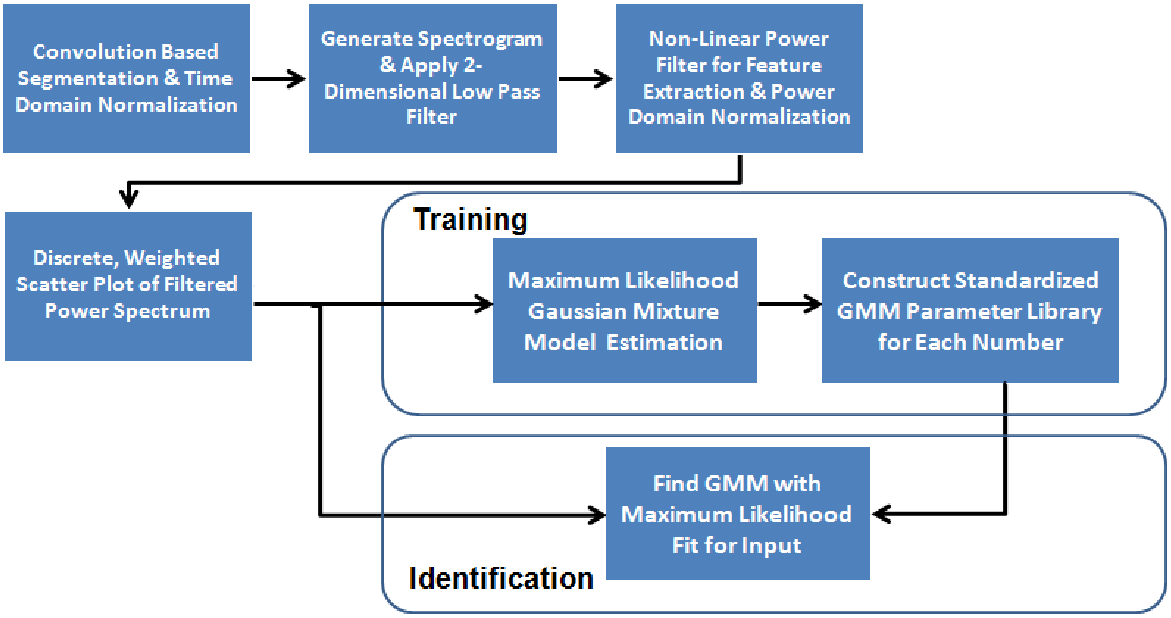
We collected the several hundreds of data samples used to train the library from ourselves.

We featured-matched input and stored data using the Yule-Walker autocorrelation method, minimizing the forward prediction error in the least squares sense. This was done using Matlab’s Yule-Walker AR Estimator.

Testing the algorithm resulted in an abysmal 20-30% accuracy.

We thought to produce better base accuracy with an algorithm of our own making. Our final results are based upon the following algorithm outlined:

1. Convolution-based segmentation
2. Feature extraction of formants via nonlinear power filter
3. Display filtered spectrum on a discrete, weighted scatter plot
4. Trace out contours of the maximum-likelihood Gaussian Mixture Model (GMM) using a maximum-likelihood GMM estimator
5. Construct a standardized GMM parameter library for each number
6. Find the GMM matching the input with a maximum-likelihood fit


Convolution Based Segmentation & Time Domain Normalization
Generate Spectrogram & Apply 2-Dimensional Low Pass Filter
Non-Linear Power Filter for Feature Extraction & Power Domain Normalization
Discrete, Weighted Scatter Plot of Filtered Power Spectrum
Training
Maximum Likelihood Gaussian Mixture Model Estimation
Construct Standardized GMM Parameter Library for Each Number
Find GMM with Maximum Likelihood Fit for Input
Identification

## 4 - Voice Recognition: Results

Results:

Figures showing the efficacy of steps 1-4 (see Methods) are displayed below.

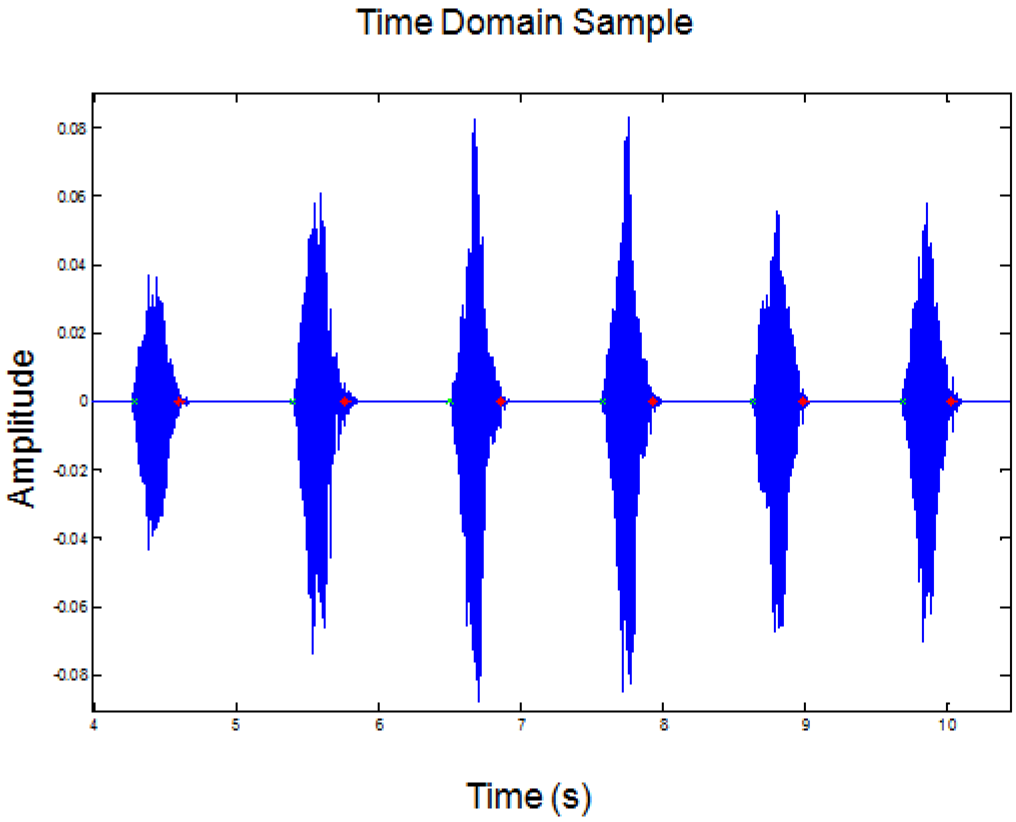
1. The signal shows clear segmentation of numbers, between the red and green markers.
2. Normalized spectrogram of an utterance of the number “1”. Formants visible but not distinct.
3. By filtering the signal with a filter to the fifth or sixth exponential order, we distinctly emphasize the difference between the formants and the background.
4. Weighted scatter plot overlaid with the contours of the maximum-likelihood GMM, showing the formants.
5. Filtered spectrum on the mel scale, with a corner frequency of 700 Hz used.
6. GMM as generated by the mel scale. Differs greatly from the linear-frequency GMM.

Testing this algorithm in Matlab with the generated input data of ten numbers resulted in a 70% accuracy match, vastly more successful than our attempt at linear prediction coding. However, while 70% is admittedly a decent result in the speech recognition field, one ought to remember that the system faces several important limitations (that were common to the LPC as well).

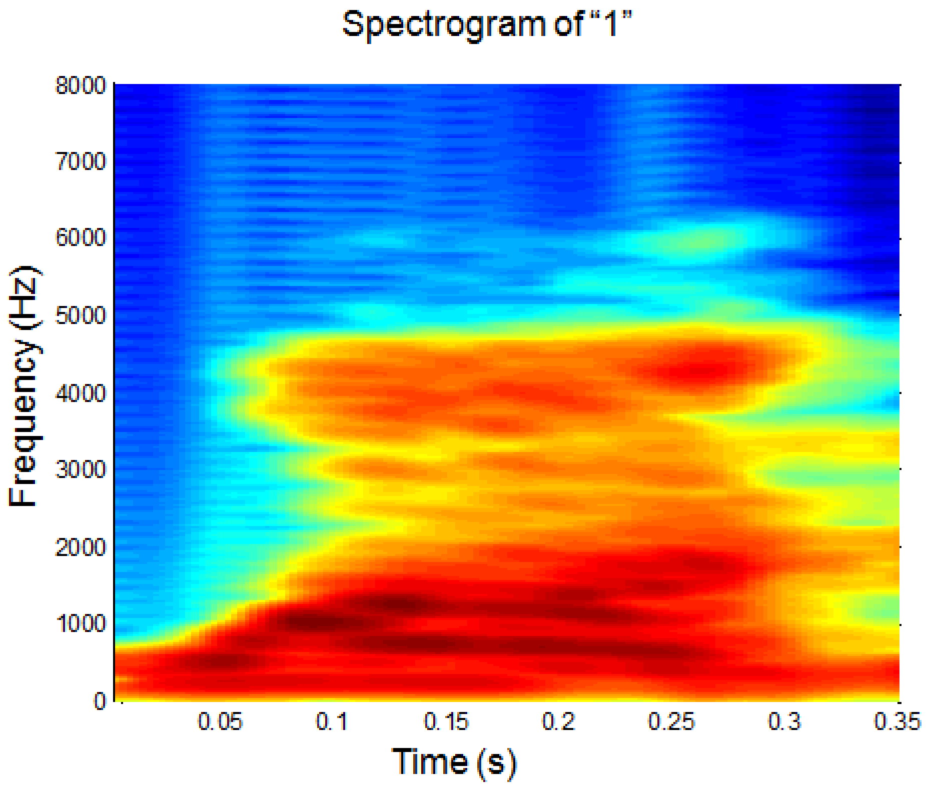
First, the system is trained by a limited sampling. While it is expected to hold to similar accuracy when tested against other male voices, it will be highly inaccurate when testing female voices. Second, segmentation has shown to work perfectly well with calm, enunciated speech, and recognition to a large degree. The same could not be said of more casual speech where numbers might be slurred or stuttered, or non-numerical noises inserted (i.e. “um” or “ah”). Similarly, some speakers might prefer to speak in terms of multiple digits - “seventy” instead of “seven-oh”, for instance. A more robust system would take these issues into account.

5 - Voice Recognition: Graphs

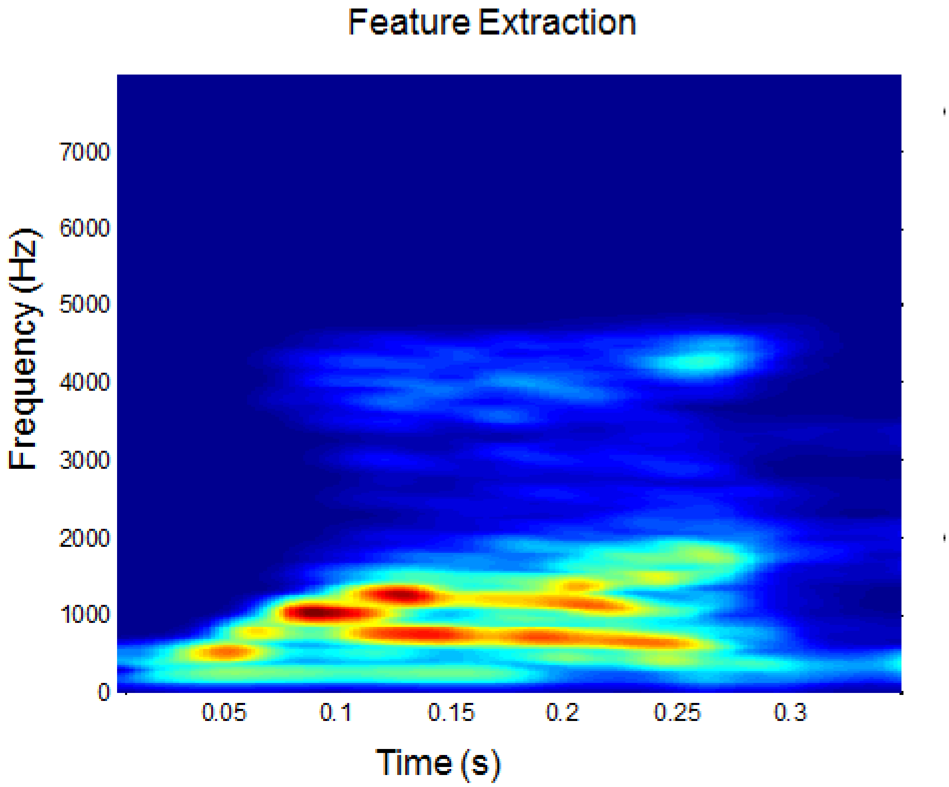
1. This time-domain signal shows clear segmentation between different numbers.



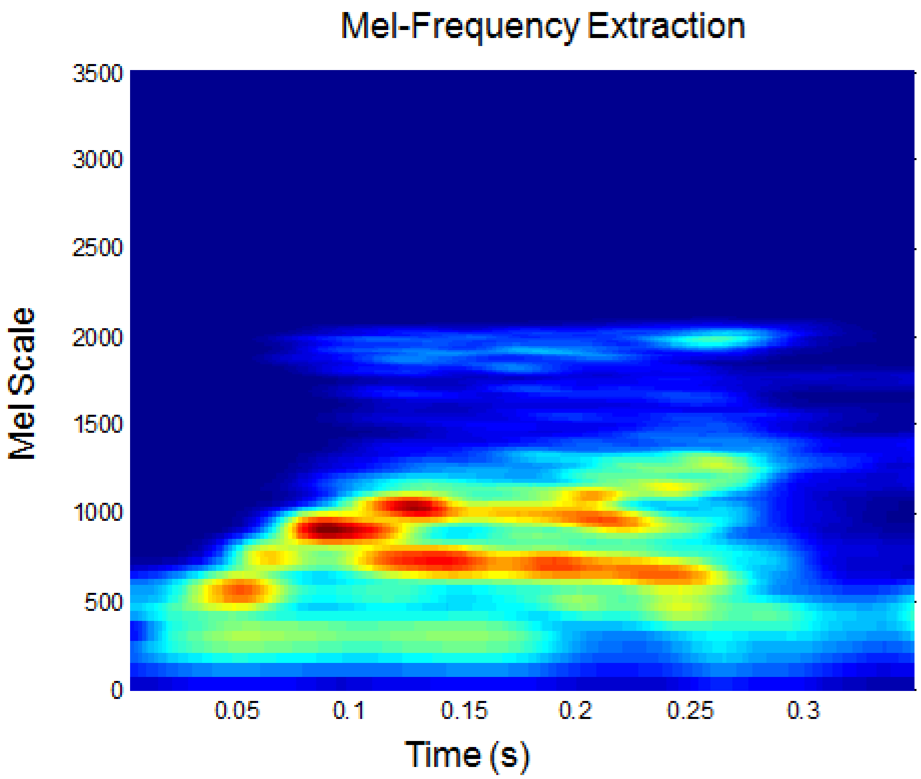
2. This shows a normalized spectrogram of the number “1”. Formants are visible but not clear.



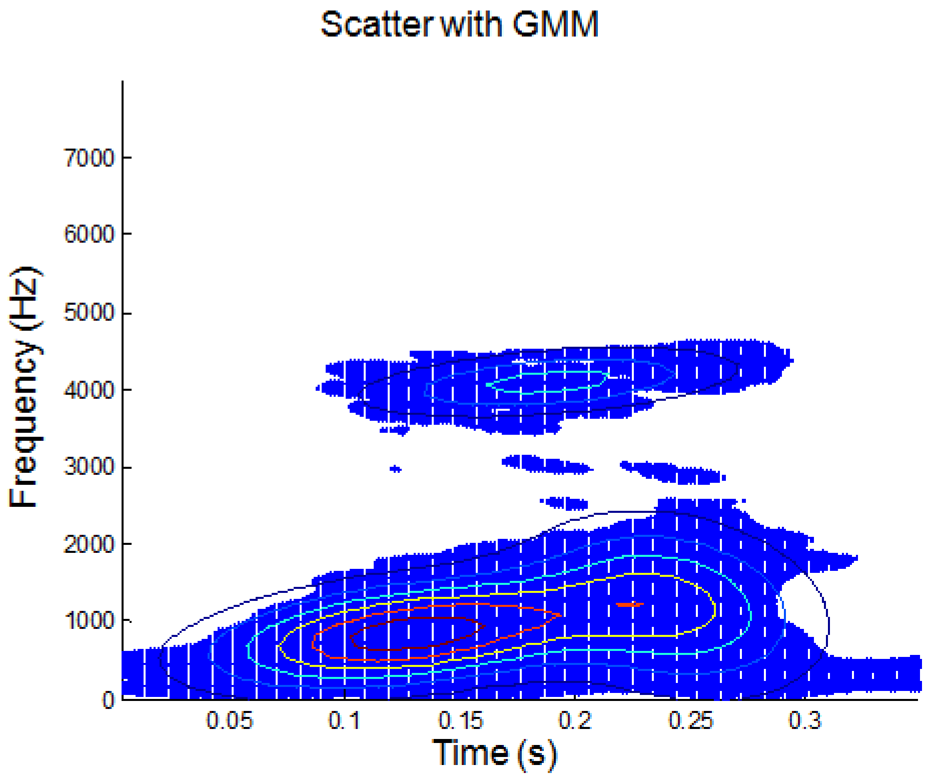
3. Enhanced by a non-linear filter to emphasize the difference between peak values and background. Formants are much more distinct.



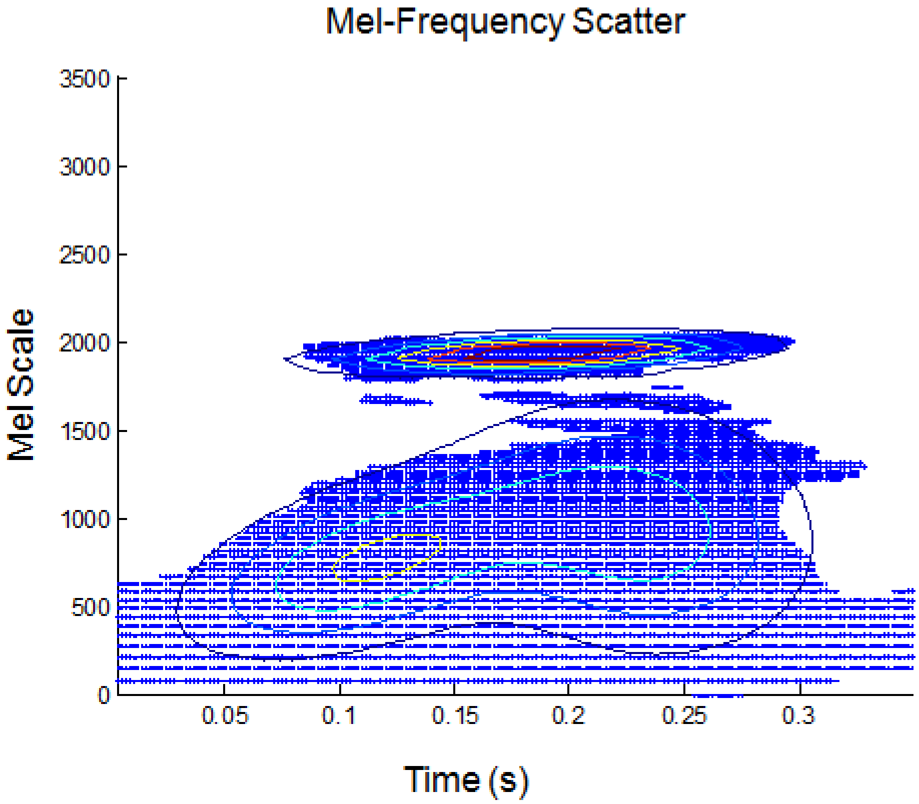
4. This shows the filtered spectrum in the Mel-scale, a logarithmic scale that models human hearing. Corner-frequency of 700 Hz used.



5. Weighted scatter plot with contours of the maximum likelihood GMM overlaid, showing the formants.



6. The GMM generated by the Mel-scale signal differs greatly from the linear-frequency version.


Mel-Frequency Scatter
Mel Scale
3500
3000
2500
2000
1500
1000
500
0
0.05
0.1
0.15
0.2
0.25
0.3
Time (s)

# 6 - Voice Recognition: Conclusions and Future Improvements

## Conclusions and Future Improvements:

The project yielded reasonably accurate results using our independent algorithm. Unfortunately, it is a severely limited method of speech recognition. Feature-matching might be improved with an optimal determination of the number of Gaussians for the GMM of each sample, and recognition of female voices might be resolved easily enough with an appropriate frequency normalization filter, but the other aforementioned issues present much more serious problems, due to the sheer number of possible variations.

A more robust speech recognition system would certainly include a larger sample size, as well as recognize based upon phonemes instead of words, as many of the modern HMM-based systems do.

Given some more time, we would have tried to salvage our attempt at the LPC as well, this time using a support vector machine for improved feature-matching.

# 7 - Voice Recognition: Acknowledgements

Acknowledgements:

We would like to thank Mr. Andrew Waters for the invaluable support he provided us on this project, as well as Professor Richard Baraniuk for sponsoring us.